DECLARATION

I, Sun YOU of 168-32, Seongbuk-gu, Seongbuk 1(il)-dong, Seoul, Republic of Korea declare that I have a through knowledge of the Korean and English language and the writings contained in the following pages are correct translations of the attached Korea Patent Applications No. 10-2000-86398 and No. 10-2002-10700.

This 18th day of August 2004

BY Sun You

Sun YOU

KOREAN INTELLECTUAL PROPERTY OFFICE

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Industrial Property Office.

Application Number: Patent Application No. 10-2000-0086398

Date of Application: December 29, 2000

Applicant(s): LG Electronics Inc.

COMMISIONER

[ABSTRACT OF THE DISCLOSURE]

[ABSTRACT]

The present invention relates to a method for controlling traffic congestion in a message transmission system, which prevents repeated bus access requests of a certain slave module within a predetermined period by providing an interval value storage in a master module..

A slave module receives authorization of bus access within a predetermined period after request of bus access. However, if there is no bus access request from other node, the slave module can directly receive authorization of bus access and transmit message. In this case, one slave module can transmit message while using all bus bandwidth. In this case, data traffic congestion may occur in the node. It is very important to know that even a single slave node having such congestion may cause the whole system to be unstable.

The present invention can prevent traffic congestion of message into a common bus by preventing repeated bus access requests of a certain slave module within a predetermined period.

[TYPICAL DRAWING]

FIG. 6

[SPECIFICATION]

[TITLE OF THE INVENTION]

METHOD FOR CONTROLLING TRAFFIC CONGESTION IN MESSAGE TRANSMISSION SYSTEM

[BRIEF DESCRIPTION OF THE DRAWINGS]

FIG. 1 illustrates a block diagram of a message transmission system of a typical private branch system;

FIG. 2 illustrates a block diagram of the master module shown in FIG. 1;

FIG. 3 illustrates a block diagram of the bus request arbiter shown in FIG. 2;

FIG. 4 illustrates a block diagram of a bust request arbiter included in a master module of the present invention;

FIG. 5 illustrates an interval value table; and

FIG. 6 illustrates a flow chart showing a bus access request process of a slave module according to the present invention.

Reference numerals of the essential parts in the drawings

400 : bus request arbiter 42 : access-authorizing block

44 : request storage 46 : interval value storage

[DETAILED DESCRIPTION OF THE INVENTION]

[OBJECT OF THE INVENTION]

[FIELD OF THE INVENTION AND DISCUSSION OF THE RELATED ART]

The present invention relates to a method for controlling traffic congestion in a message transmission system, and more particularly, to a method for controlling traffic congestion in a message transmission system, which prevent traffic congestion of message into a common bus by preventing repeated bus access requests of a certain slave module within a predetermined period.

In general, for transmitting a message from a functional module to another module in a message transmission system of a typical private branch exchange system, a bus access arbiter included in a master module arbitrates the access to bus, and the message is transmitted through a Frame Synchronous (FS) signal, which is generated by a master module. In addition, the system is able to provide both of the serial and parallel message transmission methods.

FIG. 1 illustrates a block diagram of a message transmission system of a typical private branch system, FIG. 2 illustrates a block diagram of the master module shown in FIG. 1, and FIG. 3 illustrates a block diagram of the bus request arbiter shown in FIG. 2.

As shown in FIG. 1, the message transmission system includes a master module 20 and a plurality of slave modules 30-1 ~ 30-N.

Here, the master module 20 and all of the slave modules are connected to the common bus (CB) and the control line (CL).

There are two ways to be connected to CB for a module: a serial line (SL) connection and a parallel line (PL) connection. The SL connection uses only one of the lines of CB, and the PL connection uses all the lines of CB.

The master module 20 is connected to all the CB lines, and each of the N slave modules can be connected to CB using the SL or PL connection.

The plurality of the slave modules 30-1 ~ 30-N can be connected to the CB by a serial/parallel line (S/PL). If the connection is carried out only by a serial line, there are only a serial receiving way and a serial transmitting way. However, if the connection is carried out only by a parallel line, both of serial/parallel receiving way and serial/parallel transmitting way are possible.

On the other hand, the control line (CL) includes a clock signal, a parity error signal, a busy signal, a frame synchronous signal, and many others.

The master module 20 makes a request for message transmission and is included in a node being able to send/receive a message and to arbitrate the access to CB. The master module 20 shown in FIG. 1 includes a master module 21, a CPU 22 and a shared memory 23 as shown in FIG. 2.

Here, the master module 21 includes: a message transmitting/receiving controller 21_2 for transmitting and receiving messages; a memory arbiter 21_3 for preventing data collision, which may be generated due to simultaneous access of the shared memory 23, and for reading the transmitted message from the shared memory 23 or recording it into the shared memory 23; a bus request arbiter 21_4 for determining bust access node by receiving authorization of bus access request signal of the salve modules 30-1 ~ 30-N or the master module 20, determining a transmission method after receiving request from the slave modules 30-1 ~ 30-N of a receiving side, informing it to the slave modules 30-1 ~ 30-N of a transmitting side, and generating FS signals and clock signals; and a serial/parallel controller 21-1 for converting information into serial/parallel mode between the message transmitting/receiving controller 21_2, the bust request arbiter 21-4 and the common bus CB according to the transmission mode.

Here, the bus request arbiter 21-4 includes a request storage 21-4b for storing bus access request of the slave module, and an access authorizing block 21-4a.

Reference will now be made in detail to the method of operating the message transmission system of a typical branch exchange system.

We know that only one node may use the CB for a given period of time. Therefore, a node must initially obtain a right to access to CB in order to be able to send a message to another node through CB.

Referring to the drawings, the bust request arbiter 21-3 of the master bust controller 21 included in the master module 20 receives and arbitrates a bus request

signal from the slave modules 30-1 ~ 30-N or the corresponding mater module 20, and then, one of the corresponding slave modules 30-1 ~ 30-N or the master module 20 give a right of access of the common bus.

Namely, the request storage 21-4b of the bus request arbiter 21-4 shown in FIG. 3 stores the access requesting signals, which are sent by the access requesting modules, in an order that they were received in a given period of time. After all the requesting signals are stored in the request storage 21-4b, the access authorizing block 21-4a gives an approval to a node, which is selected by using a given selecting rule.

Hereinafter, a method for requesting access of the common bus will be described as follows. The slave modules 30-1 ~ 30-N authorizes a corresponding bus access request signal to the master module 20 through the common bus (CB). At this time, since the module capable of transmitting message has an inherent node number, the bus access request signal corresponding to the 'Q' times is loaded on the common bus (CB) at 'P' clock cycles on the basis of an FS signal authorized from the master module 20.

Here, the 'P' and 'Q' has the following mathematical formula.

Mathematical formula 1

Q = the rest value in 'node number ÷ bus width'

The message transmission process will be described as follows. If message to be transmitted exists in the slave modules 30-1 ~ 30-N, when the CPU 32 records the message in the CM 33 through the slave bus controller 31, the memory arbiter 31-3 informs the message transmitting and receiving controller 31-2 that there is a message to be transmitted, and the corresponding message transmitting and receiving controller 31-2 transmits a bus access request signal to the serial/parallel controller 31-1.

The serial/parallel controller 31-1 converts the bus access request signal transmitted from the message transmitting and receiving controller 31-2 into

serial/parallel information according to the transmission mode, and transmits it to the master module 20 through the common bus (CB).

Therefore, the master bus controller 21 included in the master module 20 stores the bus access request signal transmitted through the CB, determines a bus access node in a round robin way, and then, informs the corresponding bus access node through the CB.

Furthermore, the master bus controller 21 provides a Frame Synchronous (FS) signal and a clock to the plurality of slave modules 30-1 ~ 30-N in order to provide and obtain information through the CB within an appointed time.

The plurality of the slave modules 30-1 ~ 30-N compares its own node with data provided through the CB from the master bus controller 21, and determines whether or not there is a bus access right.

If the slave modules 30-1 ~ 30-N obtains the bus access right, the slave module transmits the node number of the receiving side through the CB.

At this time, the slave modules 30-1 ~ 30-N of the other node, which do not obtain the bus access right, is in a receiving condition, compares its own node with the node number information of the receiving side transmitted from the node, which has the bus access right, and if they are the same, transmits the receiving way to the master module 20 through the CB.

The mater module 20 obtains the receiving way information from the slave modules 30-1 ~ 30-N, and informs the transmitting way to the salve modules 30-1 ~ 30-N through the CB.

Thereafter, the slave module 30-1 ~ 30-N, which has the bus access right, sends information regarding its preferred receiving method to the master module 20 through CB. Then the master module 20 informs the transmitting module of the preferred

method. Finally, the transmitting module transmits the message to the receiving module using the preferred message transfer method (i.e., serial or parallel transfer method).

However, the method of transmitting a message in a message transmission system included in a typical private branch exchange system has several disadvantages as follows.

A slave module receives authorization of bus access within a predetermined period after request of bus access. If there is only one node sending an access-requesting signal, the bus controller 21 gives an access right immediately after it receives the requesting signal. Then the node may use all the bandwidth of the bus. In this case, data traffic congestion may occur in the node. It is very important to know that even a single slave node having such congestion may cause the whole system to be unstable.

[TECHNICAL TASKS TO BE ACHIEVED BY THE INVENTION]

Accordingly, to obviate one or more problems due to limitations and disadvantages of the related art, an object of the present invention is to provide a method for preventing data traffic congestion in a message transmission system by arbitrating an access to a common bus in a manner that each module may be able to send an access requesting signal only after its previously set standby period is expired.

[PREFERRED EMBODIMENTS OF THE INVENTION]

To achieve these objects and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, a method for controlling traffic congestion in a message transmission system comprising the steps of: reading a prescaler value and an interval value stored in an interval value storage, and storing them in a memory element; down-counting the prescaler value stored in the interval value storage every predetermined cycle; determining whether the prescaler

value is equal to a first predetermined value; decreasing the interval value if the prescaler value is equal to the first predetermined value, reading the prescaler value stored in the interval value storage, storing it in the memory element, and down-counting again; determining whether the interval value is equal to a second predetermined value; and authorizing a bus access request of the corresponding slave module if the interval value is equal to the second predetermined value, and determining bus access.

Here, as a result of the determination, if the prescaler value is not equal to the first predetermined value, repeatedly down-counting the prescaler value every predetermined cycle. If the interval value stored in the interval value storage is equal to the second predetermined value, bus access is not authorized to the corresponding slave module.

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings.

FIG. 4 illustrates a block diagram of a bust request arbiter included in a master module of the present invention.

As it is shown in FIG. 4, the bus request arbiter 400 includes an access-authorizing block 42, a request storage 44, and an interval value storage 46.

The interval value storage 46 stores a prescaler value and an interval value of a slave module. Both values, which are initially set by a user for each slave module, are used to stop each slave module from re-accessing to the common bus before its given standby period is expired. For example, the interval value storage 46 contains an interval value table including an address of each slave module, the prescaler value, the interval value, a prescaler value storage, and the interval value storage. The table is shown in FIG. 5.

The access-authorizing block 42 reads the prescaler value and the interval value of each slave module stored in the interval value storage 46, and it determines whether a given standby period is expired for each salve module by down-counting prescaler value and the interval value stored in the interval value storage 46.

The request storage 44 is not described in this invention since it has the same functions as the request storage (21-4b of FIG 3) of the conventional bus request arbiter..

FIG 6 illustrates a flow chart showing a bus access request process of a slave module according to the present invention.

First, a system user sets a prescaler value and an interval value in the interval value storage 42 (S61). If the interval value of a slave module is set to zero, the access-authorizing block 42 will not give an access right to the slave module.

Next, the access-authorizing block 42 reads the prescaler and interval values of the slave module and stores the values in a memory included in a built-in memory. And the block 42 down-counts the stored prescaler value for an every given period (S62).

Thereafter, it checks whether the down-counted prescaler value is zero (S63), and it repeats the step S62 until the down-counted prescaler value becomes zero. When it is finally determined in the step S63 that the prescaler value is zero, the block 42 down-counts the interval value initially stored in the built-memory (S64).

Next, the access-authorizing block 42 checks whether the down-counted interval value is zero (65). If it is not, the access-authorizing block 42 returns to the step S62 to decrease the prescaler and interval values. However, if the interval value is zero, the access-authorizing block 42 determines availability of the bus access request of the corresponding slave module, and arbitrates bus access by applying a typical round robin method (S66). After that, if the bus access authorization on the corresponding slave module is determined according to the determination of the access-authorizing

block 42 (S67), the corresponding slave module can transmit message through the bus (S68).

It will be apparent to those skilled in the art than various modifications and variations can be made in the present invention. Thus, it is intended that the present invention covers the modifications and variations of this invention provided they come within the scope of the appended claims and their equivalents.

[EFFECT OF THE INVENTION]

As described above, the present invention can prevent traffic congestion of message into a common bus by preventing repeated bus access requests of a certain slave module within a predetermined period.

What Is Claimed Is:

1. A method for controlling traffic congestion in a message transmission system comprising the steps of:

reading a prescaler value and an interval value stored in an interval value storage, and storing them in a memory element;

down-counting the prescaler value stored in the interval value storage every predetermined cycle;

determining whether the prescaler value is equal to a first predetermined value;

decreasing the interval value if the prescaler value is equal to the first predetermined value, reading the prescaler value stored in the interval value storage, storing it in the memory element, and down-counting again;

determining whether the interval value is equal to a second predetermined value; and

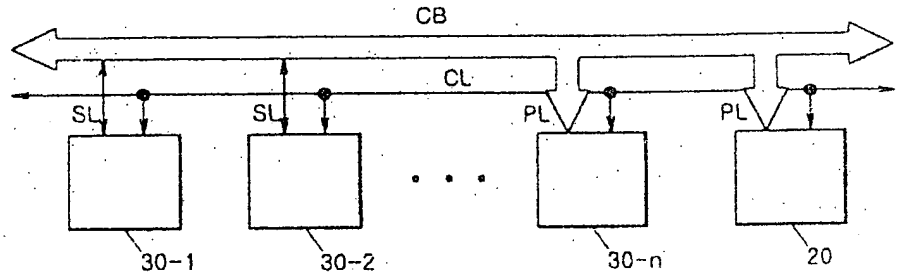
authorizing a bus access request of the corresponding slave module if the interval value is equal to the second predetermined value, and determining bus access.

2. The method for controlling traffic congestion in a message transmission system as claimed in claim 1, wherein as a result of the determination, if the prescaler value is not equal to the first predetermined value, repeatedly down-counting the prescaler value every predetermined cycle.

3. The method for controlling traffic congestion in a message transmission system as claimed in claim 1, wherein if the interval value stored in the interval value storage is equal to the second predetermined value, bus access is not authorized to the

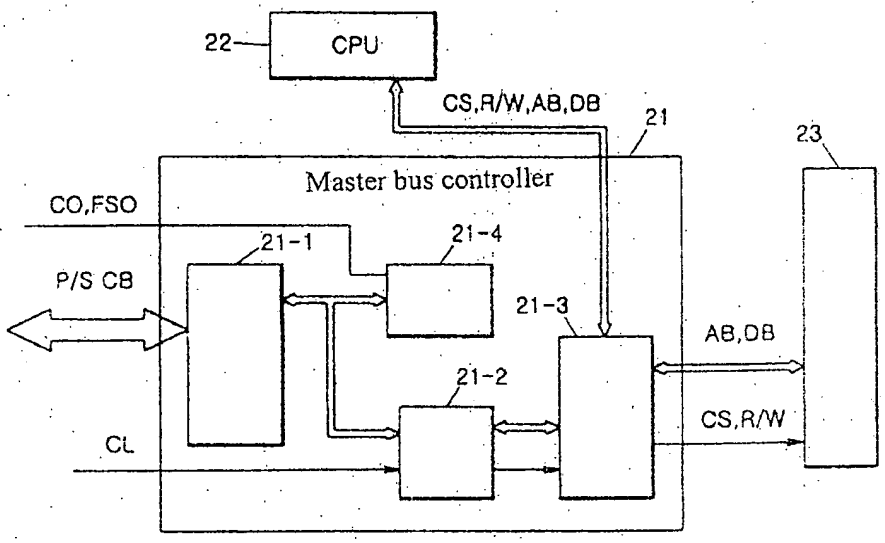
corresponding slave module.

FIG. 1



1st slave module, 2nd slave module nth slave module, master module

FIG. 2



21-1: serial/parallel controller

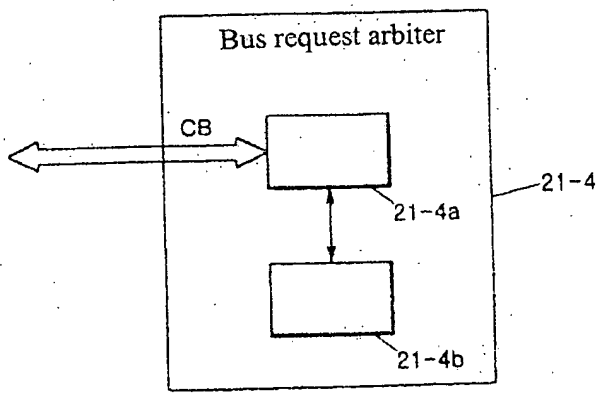
21-2: message transmitting and receiving controller

21-3: memory arbiter

21-4: bus request arbiter

23: shared memory

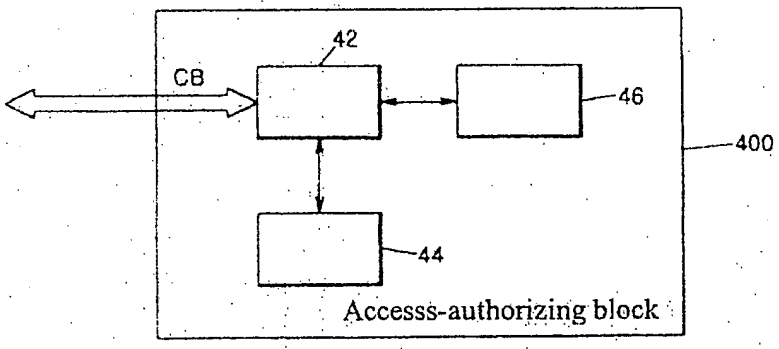
FIG. 3



21-4a: access-authorizing block

21-4b: request storage

FIG. 4



44: request storage

46: interval value storage

FIG.5

Address	prescaler value	interval value	prescaler value storage	interval value storage
0	X	X	X	X
1	X	X	X	X
2	X	X	X	X
n	X	X	X	X

FIG. 6